

# Cloudflare Dynamic DNS

Do you have your own domain on Cloudflare and want to update it like you would with a dynamic DNS service, but don't want to use said Dynamic DNS service?

Then this guide is for you!

What you'll need for these scripts to work for you:

- VPS or similar sitting in the cloud or at a remote location that is running a web server and PHP.
- Composer installed.
- Local VPS, server or anything that runs Linux with Cron and Curl.
- Already configured Cloudflare domain with a new sub domain setup for the dynamic DNS.
- Redis installed.

First of all we need to grab the required API so do a `composer require cloudflare/sdk`

Afterwards setup the following PHP script on your server along with the vendor folder you just downloaded.

```
<?php
$token = $_REQUEST["token"];
if ($token != "<some random string>")
    exit("Access denied.");
require_once __DIR__ . "/vendor/autoload.php";
$redis = new Redis();
$redis->connect("127.0.0.1", "6379");
$lastIP = $redis->get("dynamic:last-ip");
if (!$lastIP) $lastIP = "";
$m = "";
$skip = false;
$remoteIP = $_SERVER['REMOTE_ADDR'];
if (!filter_var($remoteIP, FILTER_VALIDATE_IP)) {
    $skip = true;
    $m .= "(Remote IP invalid)";
}
```

```

if (!$skip && $lastIP != "") {
    if ($remoteIP == $lastIP) $skip = true;
    $m .= "(LastIP and RemoteIP are the same)";
}
$redis->set("dynamic:last-ip", $remoteIP);
if (!$skip) {
    $key = new Cloudflare\API\Auth\APIKey('<email>', '<cloudflare API key>'); $adapter = new
    Cloudflare\API\Adapter\Guzzle($key);
    $zones = new \Cloudflare\API\Endpoints\Zones($adapter); $zoneID = $zones->getZoneID("<root domain
    name e.g example.com>");
    $dns = new \Cloudflare\API\Endpoints\DNS($adapter); $record = $dns->getRecordID($zoneID, "A",
    "<domain plus sub domain e.g: home.example.com>");
    $dns->updateRecordDetails($zoneID, $record, ["type" => "A", "name" => "<domain plus sub domain
    e.g: home.example.com>", "content" => $remoteIP]);
}
if (!$skip) {
    echo "done";
} else {
    echo "Skipped: ".$m;
}
}

```

You will see from the script that I'm using Redis to easily identify if I've already tried updating with the same IP address twice. This stops me flooding Cloudflare with useless requests but it can be removed if desired.

Next up on our home server we'll make a new Cron script to call the script we just made on the remote server. Do a `crontab -e` and add:

```
* * * * * curl -d "token=<your random string from above>" -X POST <url to the webserver running the
```

Now your DNS will be updated automatically every time it changes.

Enjoy!

---

Revision #4

Created Wed, Aug 7, 2019 7:18 PM by Kore

Updated Thu, Sep 3, 2020 7:48 AM by Kore